On Algebraic Decoding of q-ary Reed-Muller and Product Reed-Solomon Codes

Nandakishore Santhi[*]
Theoretical Division, CCS-3 Division and the Center for Non Linear Studies
LANL, MS B213, T-13, Los Alamos, NM 87545
nsanthi@lanl.gov

***Abstract*— We consider a list decoding algorithm recently proposed by Pellikaan-Wu [8] for q-ary Reed-Muller codes $\mathscr{RM}_q(\ell,\, m,\, n)$ of length $n \le q^m$ when $\ell \le q$. A simple and easily accessible correctness proof is given which shows that this algorithm achieves a relative error-correction radius of $\tau \le \left(1 - \sqrt{\ell q^{m-1}/n}\right)$. This is an improvement over the proof using one-point Algebraic-Geometric codes given in [8]. The described algorithm can be adapted to decode Product-Reed-Solomon codes.**

We then propose a new low complexity recursive algebraic decoding algorithm for Reed-Muller and Product-Reed-Solomon codes. Our algorithm achieves a relative error correction radius of $\tau \le \prod_{i=1}^{m}\left(1 - \sqrt{k_i/q}\right)$. This technique is then proved to outperform the Pellikaan-Wu method in both complexity and error correction radius over a wide range of code rates.

I. Introduction

With the discovery of deterministic list-decoding algorithms for several Algebraic-Geometric codes, most notably the Guruswami-Sudan [6] algorithm, there has been renewed interest in algebraic decoding methods for other related q-ary codes such as the Reed-Muller [7], [8] and Product-Reed-Solomon [9] codes. However some of the existing correctness proofs for these algorithms use advanced algebraic geometric tools. In this paper we first derive a proof for a list decoding algorithm for a q-ary Reed-Muller code. Our proof is from first principles and require only the most basic notions from finite field theory. We then proceed to propose new recursive list decoding algorithms for Reed-Muller and Product-Reed-Solomon codes. These algorithms are rigorously shown to outperform the Pellikaan-Wu method in both complexity as well as error-correction-radius.

The basic idea of our new proof for the Pellikaan-Wu algorithm is to "lift" a multivariate polynomial in $\mathbb{F}_q[x_1, x_2, \ldots, x_m]$ to a univariate polynomial in $\mathbb{F}_{q^m}[X]$ using a deterministic mapping rule. This in turn results in a higher total degree polynomial. The increase in degree will not be high enough to render our list decoding strategy for Reed-Muller codes useless at meaningful rates. A higher degree for the lifted polynomial means that this Reed-Muller code list decoding algorithm has a lower relative error-correction radius (as a function of the rate) than a comparable rate Reed-Solomon list decoder based on the Guruswami-Sudan algorithm. In the following section we describe the mapping rule and the decoding algorithm in some detail.

In the final section we propose new algorithms for decoding Reed-Muller and Product-Reed-Solomon codes. Our algorithm is more efficient than the Pellikaan-Wu method by approximately a quadratic factor. Furthermore it outperforms the Pellikaan-Wu algorithm in error-correction-radius over a wide range of code rates.

[*] Document: LA-UR-07-0469.

II. Correctness of a List Decoding Algorithm

Let us begin by defining a q-ary Reed-Muller code.

Definition 1 *The q-ary Reed-Muller code $\mathscr{RM}_q(\ell,\, m,\, n)$ of length $n \le q^m$ is defined as the set of vectors given by:*

$$\mathscr{RM}_q(\ell,\, m,\, n) \overset{\text{def}}{=} \{\; [\varphi(\boldsymbol{\alpha}_1)\,\varphi(\boldsymbol{\alpha}_2)\,\cdots\,\varphi(\boldsymbol{\alpha}_n)] \;\mid\; \varphi \in \mathbb{F}_q[x_1, x_2, \ldots, x_m],\ \deg(\varphi) \le \ell \;\} \tag{1}$$

where $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_n\}$ are any set of n distinct points in $\mathbb{F}_q^m$. Here by $\deg(\varphi)$ we mean the total degree of the multivariate polynomial φ.

The following well known property will be useful:

Proposition 1 *Let $\{a_1, a_2, \ldots, a_m\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and let $[x_1 x_2 \ldots x_m] \in \mathbb{F}_q^m$. Then the map $\psi : \mathbb{F}_q^m \to \mathbb{F}_{q^m}$ defined as in (2) is an isomorphism.*

$$[x_1 x_2 \ldots x_m] \;\mapsto\; X \overset{\text{def}}{=} \sum_{j=1}^{m} a_j x_j \tag{2}$$

For example one might as usual use a polynomial basis $\{1, \xi, \xi^2, \ldots, \xi^{m-1}\}$ where ξ is any primitive element in $\mathbb{F}_{q^m}$ or even a normal basis of the form $\{\zeta, \zeta^q, \zeta^{q^2}, \ldots, \zeta^{q^{m-1}}\}$, where ζ is a suitable primitive element in $\mathbb{F}_{q^m}$.

Therefore we arrive at this elementary conclusion:

Lemma 1 *Let $X \in \mathbb{F}_{q^m}$. The reverse isomorphism for (2) is:*

$$X \;\mapsto\; [x_1 x_2 \ldots x_m]^T \overset{\text{def}}{=} \boldsymbol{A}^{-1} \cdot [X\, X^q\, X^{q^2}\, \ldots\, X^{q^{m-1}}]^T \tag{3}$$

where

$$\boldsymbol{A} \overset{\text{def}}{=} \begin{bmatrix} a_1 & a_2 & \ldots & a_m \\ a_1^q & a_2^q & \ldots & a_m^q \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{q^{m-1}} & a_2^{q^{m-1}} & \ldots & a_m^{q^{m-1}} \end{bmatrix} \tag{4}$$

is a non-singular (invertible) square matrix.

Proof: Since $X = \sum_{j=1}^{m} a_j x_j$, and $x_j \in \mathbb{F}_q$, we get $X^{q^i} = \sum_{j=1}^{m} a_j^{q^i} x_j$ using Fermat's little theorem. It only remains to show that $\boldsymbol{A}$ is non-singular. Note that in general $\boldsymbol{A}$ is *not* a Vandermonde matrix. However by construction, the set $\{a_1, a_2, \ldots, a_m\}$ is a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. It then follows from [4, Corollary 2.38, pp. 58] that $\boldsymbol{A}$ is non-singular. ■

It follows from Lemma 1 that there exist polynomials $\mu_j \in \mathbb{F}_{q^m}[X]$ of degree at most q^{m-1} such that $x_j = \mu_j(X), 1 \le j \le m$. Substituting for all x_j in this manner, we have proved the following:

Theorem 1 *Let $n \le q^m$. If $\ell \le q$ then*

$$\mathscr{RM}_q(\ell,\, m,\, n) \subseteq \mathscr{RS}_{q^m}(n,\, \ell q^{m-1}) \cap \mathbb{F}_q^n \tag{5}$$

where $\mathscr{RS}_{q^m}(n,\, \ell q^{m-1})$ is the Reed-Solomon code given by

$$\begin{aligned}\mathscr{RS}_{q^m}(n,\, \ell q^{m-1}) \overset{\text{def}}{=} \{ \; & [f(\beta_1)\, f(\beta_2) \,\ldots\, f(\beta_n)] \\ & \mid \; f \in \mathbb{F}_{q^m}[X],\, \deg(f) \le \ell q^{m-1} \; \}\end{aligned} \tag{6}$$

where $\beta_i \overset{\text{def}}{=} \sum_{j=1}^{m} a_j \alpha_{ij}$, and $\boldsymbol{\alpha}_i \overset{\text{def}}{=} [\alpha_{i1}\ \alpha_{i2}\ \ldots\ \alpha_{im}]$, $1 \le i \le n$ are the points of evaluation for the Reed-Muller code. Moreover if the information polynomial associated with the Reed-Muller code is given by

$$\varphi(x_1, x_2, \ldots, x_m) \overset{\text{def}}{=} \sum_{\substack{i_1, i_2, \ldots, i_m: \\ \sum_j i_j \le \ell}} \varphi_{i_1, i_2, \ldots, i_m} \prod_{j=1}^{m} x_j^{i_j} \tag{7}$$

then the information polynomial f of degree at most ℓq^{m-1} associated with the Reed-Solomon code is:

$$f(X) = \sum_{\substack{i_1, i_2, \ldots, i_m: \\ \sum_j i_j \le \ell}} \varphi_{i_1, i_2, \ldots, i_m} \prod_{j=1}^{m} (\mu_j(X))^{i_j} \tag{8}$$

Let $d_H(\boldsymbol{x},\, \boldsymbol{y})$ represent the Hamming distance between the two vectors. Using Theorem 1 and the Guruswami-Sudan algorithm [6] for list decoding a Reed-Solomon code, we have proved the correctness of the following deterministic list-decoding algorithm for Reed-Muller codes:

Algorithm 1 (RM-List-1)

```
INPUT: $q, \ell \le q, m, n \le q^m$; $\boldsymbol{r} = [r_1\ r_2\ \ldots\ r_n] \in \mathbb{F}_q^n$.
STEPS:
1. Compute the parameter $t = \left\lceil n\left(1 - \sqrt{\ell q^{m-1}/n}\right)\right\rceil$.
2. Using Guruswami-Sudan algorithm find a list $\mathscr{L}$ of code-
   words $\boldsymbol{c} \in \mathscr{RS}_{q^m}(n,\, \ell q^{m-1})$ such that $d_H(\boldsymbol{c},\, \boldsymbol{r}) < t$.
3. For every $\boldsymbol{c} \in \mathscr{L}$ check if $\boldsymbol{c} \in \mathbb{F}_q^n$ :
     i. If NO then discard $\boldsymbol{c}$ from $\mathscr{L}$.
     ii. If YES then check if $\boldsymbol{c} \in \mathscr{RM}_q(\ell,\, m,\, n)$ :
         a. If NO then discard $\boldsymbol{c}$ from $\mathscr{L}$.
         b. If YES then keep $\boldsymbol{c}$ in the list $\mathscr{L}$.
4. return
OUTPUT: $\mathscr{L}$
```

This algorithm was originally proposed by Pellikaan-Wu in [8], though their proofs were different.

A. Complexity of Algorithm 1

The complexity of our proposed algorithm is of the same order as the complexity of Guruswami-Sudan algorithm for decoding Reed-Solomon codes over the extension field $\mathbb{F}_{q^m}$. This is $\mathscr{O}(n^3)$ field operations in $\mathbb{F}_{q^m}$.

B. Comparison to previous results

The Pellikaan-Wu algorithm for decoding Reed-Muller codes by means of embedding into one-point Algebraic-Geometric codes was shown [8] to achieve an error correction radius of $\left\lceil n\left(1 - \sqrt{\ell(q+1)^{m-1}/n}\right)\right\rceil$. It is interesting to note that the error-correction radius demonstrated herein is always larger than that suggested by the Pellikaan-Wu formalism employing Algebraic-Geometric codes. However we believe that the more important contribution of this paper is the readily accessible correctness proof which relies on just a few basic notions from Galois theory.

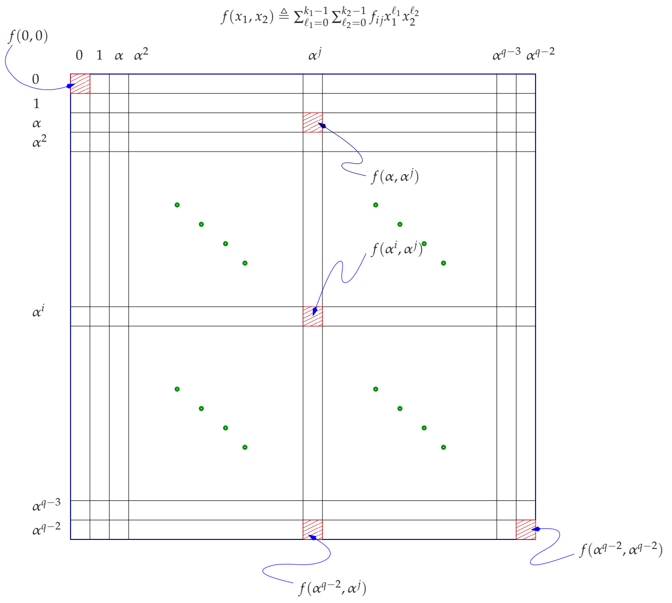
C. Product Reed-Solomon codes

Product Reed-Solomon codes $\mathscr{PRS}_{q,m}(q^m,\, k_1, \ldots, k_m) \overset{\text{def}}{=} \otimes_{i=1}^{m} \mathscr{RS}_q(q,\, k_i)$ over $\mathbb{F}_q^m$ can be thought of as the set of vectors whose q^m coordinates consist of the q^m evaluations of m-variate information polynomials with coefficients in $\mathbb{F}_q$ and degree in the i^{th}-variable x_i at most $(k_i - 1)$. m is usually called the dimension of the product code. Thus $\mathscr{PRS}_{q,m}(q^m,\, k_1, \ldots, k_m)$ is contained in $\mathscr{RM}_q(\sum_{i=1}^{m}(k_i - 1),\, m,\, q^m)$. When $\sum_{i=1}^{m}(k_i - 1) \le q$ the list decoding algorithm given in Algorithm 1 may be used essentially without any modifications. Several Product-Reed-Solomon algebraic list decoders, including a similar method as sketched above are described in [9]. Using Algorithm 1 it is possible to achieve a relative error correction radius of $(1 - \sqrt{\sum_{i=1}^{m} \rho_i})$, where $\rho_i \overset{\text{def}}{=} k_i/q$.

D. Zeros of Multivariate Polynomials

From Theorem 1, it is clear that $f(X)$ being of degree at most ℓq^{m-1}, has at most ℓq^{m-1} zeros in $\mathbb{F}_{q^m}$, including multiplicities. Therefore a non-zero multivariate polynomial $\varphi(x_1, x_2, \ldots, x_m)$ of total degree ℓ has at most ℓq^{m-1} zeros in $\mathbb{F}_q^m$. This gives the famous DeMillo-Lipton-Schwartz-Zippel[2] lemma for polynomials over finite fields. Note that the statement above appears to be stronger than the classical lemma in that this counts multiplicities too. Moreover the proof also appears to differ from the traditional expositions which use probabilistic arguments.

Next we propose a lower complexity recursive algebraic decoder which outperforms the Reed-Muller decoder considered in this section.



$f(x_1,x_2) \triangleq \sum_{\ell_1=0}^{k_1-1}\sum_{\ell_2=0}^{k_2-1} f_{ij}x_1^{\ell_1}x_2^{\ell_2}$
$f(0,0)$
0 1 α α^2 α^j α^{q-3} α^{q-2}
0 1 α α^2 α^i α^{q-3} α^{q-2}
$f(\alpha,\alpha^j)$
$f(\alpha^i,\alpha^j)$
$f(\alpha^{q-2},\alpha^j)$
$f(\alpha^{q-2},\alpha^{q-2})$

Fig. 1: 2D PRS code described on a rectangular array.

III. A Recursive Decoding Algorithm for Reed-Muller and Product Reed-Solomon Codes

For simplicity, let $n \overset{\text{def}}{=} q^m$. A codeword in the code $\mathscr{PRS}_{q,m}(q^m,\, k_1,\, \ldots,\, k_m)$ can be described within an m-dimensional cube of side length q. See Figure 1. Let a codeword $\boldsymbol{c}$ (correspondingly a received word, $\boldsymbol{r}$) be so described. We will find it convenient to write this vector as $\langle\!\langle \boldsymbol{c}_{i_1,i_2,\ldots,i_m} \rangle\!\rangle$, where each of the indices i_j take values in the range $\{1,\ldots,q\}$. We further use the notation $\langle\!\langle \boldsymbol{c}^{a_j,a_{j+1},\ldots,a_m}_{i_1,i_2,\ldots,i_{j-1}} \rangle\!\rangle$ to denote the $(j-1)$-dimensional vector formed out of $\langle\!\langle \boldsymbol{c}_{i_1,i_2,\ldots,i_m} \rangle\!\rangle$ when the coordinates indexed by $(i_j, i_{j+1}, \ldots, i_m)$ are fixed at $(a_j, a_{j+1}, \ldots, a_m)$ and the rest of the indices are free. By the nature of the product code, $\langle\!\langle \boldsymbol{c}^{a_j,a_{j+1},\ldots,a_m}_{i_1,i_2,\ldots,i_{j-1}} \rangle\!\rangle$ belongs to $\mathscr{PRS}_{q,j-1}(q^{j-1},\, k_1,\, \ldots,\, k_{j-1})$.

Now consider the following decoding algorithm for the code $\mathscr{PRS}_{q,m}(q^m,\, k_1,\, \ldots,\, k_m)$:

Algorithm 2 (PRS-Decoder)

```
INPUT: $q, (k_1,k_2,\ldots,k_m) : k_i < q, m;\ \boldsymbol{r} \in \mathbb{F}_q^n$, where $\boldsymbol{r} \overset{\text{def}}{=} \langle\!\langle r_{i_1,i_2,\ldots,i_m} \rangle\!\rangle; 1 \le i_j \le q$.
STEPS:
1. If $m = 1$ do:
    i. Compute the parameter $t_1 = \left\lceil q\left(1 - \sqrt{k_1/q}\right)\right\rceil$.
    ii. Using Guruswami-Sudan algorithm find a list $\mathscr{L}_1$ of
        codewords $\boldsymbol{c}_1 \in \mathscr{RS}_q(q,\, k_1)$ such that $d_H(\boldsymbol{c}_1,\, \langle\!\langle \boldsymbol{r}_{i_1} \rangle\!\rangle) <$
        $t_1$.
    iii. Search $\mathscr{L}_1$ for $\boldsymbol{c}_1$ such that $d_H(\boldsymbol{c}_1,\, \langle\!\langle \boldsymbol{r}_{i_1} \rangle\!\rangle)$ is least.
         Substitute in-place the positions corresponding to $\langle\!\langle \boldsymbol{r}_{i_1} \rangle\!\rangle$
         in $\boldsymbol{r}$ with $\boldsymbol{c}_1$ and return.
2. For $a_m = 1, 2, \ldots, q$ do:
    i. Set $\boldsymbol{r}' \leftarrow \langle\!\langle \boldsymbol{r}^{a_m}_{i_1,i_2,\ldots,i_{m-1}} \rangle\!\rangle$
    ii. Set $m' \leftarrow m-1$ and $n' \leftarrow q^{m'}$
    iii. Recursively decode $\boldsymbol{r}'$ using PRS-Decoder with input
         parameters $q, (k_1,k_2,\ldots,k_{m'}), m';\ \boldsymbol{r}' \in \mathbb{F}_q^{n'}$.
3. Compute the parameter $t_m = \left\lceil q\left(1 - \sqrt{k_m/q}\right)\right\rceil$.
4. For each $m-1$ tuple $(a_1, a_2, \ldots, a_{m-1})$ do:
    i. Using Guruswami-Sudan algorithm find a list
       $\mathscr{L}_m$ of codewords $\boldsymbol{c}_m \in \mathscr{RS}_q(q,\, k_m)$ such that
       $d_H(\boldsymbol{c}_m,\, \langle\!\langle \boldsymbol{r}^{a_1,a_2,\ldots,a_{m-1}}_{i_m} \rangle\!\rangle) < t_m$.
    ii. Search $\mathscr{L}_m$ for $\boldsymbol{c}_m$ such that $d_H(\boldsymbol{c}_m,\, \langle\!\langle \boldsymbol{r}^{a_1,a_2,\ldots,a_{m-1}}_{i_m} \rangle\!\rangle)$ is
        least. Substitute in-place the positions corresponding
        to $\langle\!\langle \boldsymbol{r}^{a_1,a_2,\ldots,a_{m-1}}_{i_m} \rangle\!\rangle$ with $\boldsymbol{c}_m$.
5. return
OUTPUT: Resulting vector $\boldsymbol{r}$
```

The following recursive algorithm uses **PRS-Decoder** to decode $\mathscr{RM}_q(\ell,\, m,\, n)$.

Algorithm 3 (RM-List-2)

```
INPUT: $q, \ell \le q, m, n \le q^m;\ \boldsymbol{r} = [r_1\ r_2\ \ldots\ r_n] \in \mathbb{F}_q^n$.
STEPS:
1. For each possible m-tuple $(k_1,k_2,\ldots,k_m) : k_i < q, \sum_j k_j \le \ell$
   do:
    i. Using PRS-Decoder with input parameters
       $q, (k_1,k_2,\ldots,k_m), m;\ \boldsymbol{r} \in \mathbb{F}_q^n$, decode $\boldsymbol{r}$ as $\boldsymbol{c}$.
    ii. Add $\boldsymbol{c}$ to a list $\mathscr{L}$ of codeword candidates.
2. return
OUTPUT: $\mathscr{L}$
```

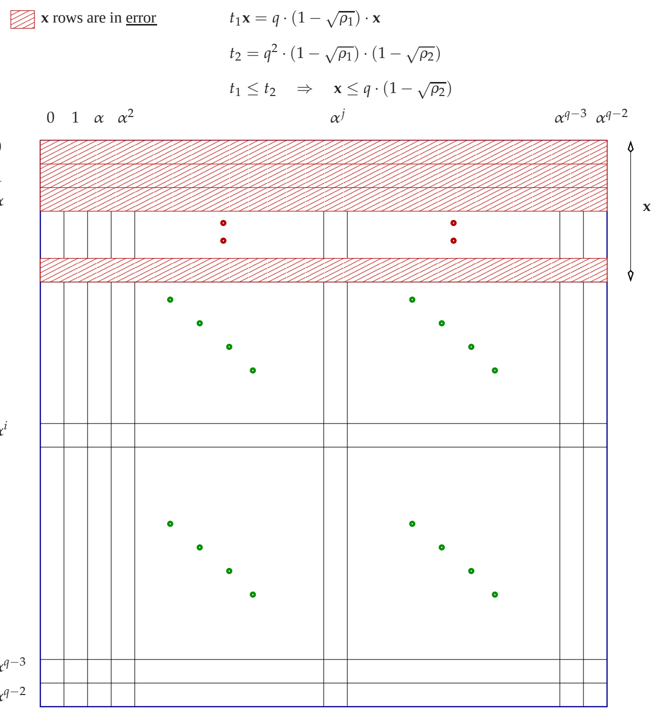
We have the following result concerning the decoding power of Algorithm 2 and Algorithm 3.

Theorem 2 *Algorithm 2 has a relative error correction radius of $\tau_m \overset{\text{def}}{=} \prod_{i=1}^m (1-\sqrt{\rho_i})$, where $\rho_i \overset{\text{def}}{=} k_i/q$. Moreover, there exist error patterns of weight above $n\prod_{i=1}^m (1-\sqrt{\rho_i})$ which cannot be guaranteed to be efficiently decoded by Algorithm 2.*

Proof:

Our proof is by induction. When $m = 1$, the claim is trivially true. Let us assume the claim to be true for some $m = M$. We will now show it to be true for the case $m = M+1$. Let there be a maximum of $t_{M+1} = q^{M+1}\prod_{i=1}^{M+1}(1-\sqrt{\rho_i})$ errors. In Step 2 of Algorithm 2, let there be a maximum of x recursions which fail to decode correctly. Since by the induction hypothesis, this would mean that there are more than t_M errors in these x sub-recursions, we have that $x t_M \le t_{M+1}$. Substituting for t_{M+1} and t_M gives, $x \le q(1-\sqrt{\rho_{M+1}})$. These errors will get corrected in Step 4 of the algorithm. This proves the first part of the claim. For the 2D case, the proof is concisely depicted in Figure 2.

To see the second part of the claim, we observe that an error pattern which is contiguously spread over an m dimensional sub-cube of volume more than $n\prod_{i=1}^m (1-\sqrt{\rho_i})$ cannot be guaranteed to be efficiently decoded by the proposed algorithm. This shows that the error correction radius predicted in the first part of Theorem 2 is rather tight. ■



x rows are in error
$t_1 \mathbf{x} = q \cdot (1-\sqrt{\rho_1}) \cdot \mathbf{x}$
$t_2 = q^2 \cdot (1-\sqrt{\rho_1}) \cdot (1-\sqrt{\rho_2})$
$t_1 \le t_2 \Rightarrow \mathbf{x} \le q \cdot (1-\sqrt{\rho_2})$
0 1 α α^2 α^j α^{q-3} α^{q-2}
0 1 α α^i α^{q-3} α^{q-2}
x

Fig. 2: The proof of Theorem 2 for a 2D PRS code.

A. *Complexity of Algorithm 2 and Algorithm 3*

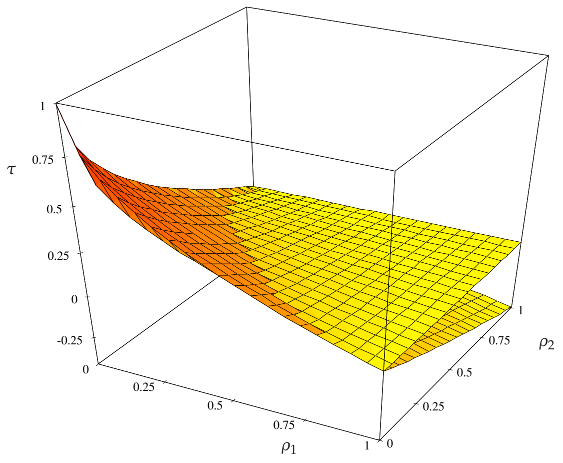
Let ϑ_m be the complexity of decoding an m-dimensional Product-Reed-Solomon code using Algorithm 2. Then the complexity of decoding an $m+1$ dimensional code is $\vartheta_{m+1} = \mathcal{O}(q\,\vartheta_m + q^m\,\vartheta_1)$. But $\vartheta_1 = \mathcal{O}(q^3)$ field operations in $\mathbb{F}_q$. This gives, $\vartheta_m = \mathcal{O}(q^{m+2})$ which is $\approx \mathcal{O}(n)$ for large m. The complexity of Algorithm 3 is $\approx \mathcal{O}(n^2)$ field operations in $\mathbb{F}_q$. This is substantially better than the Pellikaan-Wu method in Algorithm 1.

B. *Comparison of Algorithm 1 and Algorithm 3*

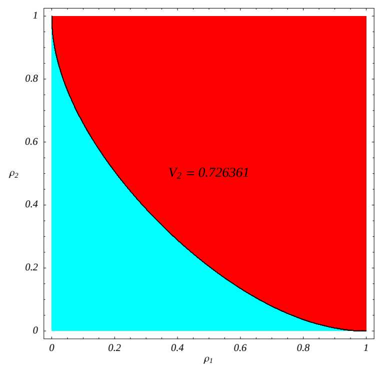
Algorithm 3 not only has a lower complexity, but also performs better over a wide range of rates. For example when $\sum_i \rho_i > 1$, the Pellikaan-Wu algorithm is not effective, whereas the new algorithm is still useful. Furthermore $\prod_{i=1}^{m}(1-\sqrt{\rho_i})$ is larger than $(1-\sqrt{\sum_{i=1}^{m}\rho_i})$ for most code rates and the advantage is more pronounced at higher code rates. Figure 3 shows the decoding power of Algorithm 2.

C. *Other Related Product Code Decoders*

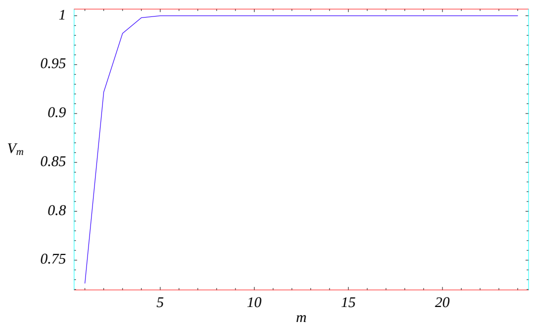
Several iterative hard decision decoders for Product-Reed-Solomon codes available in literature use some form of Algorithm 2. Usually such algorithms are described with no theoretical bounds on their error correction radii. These product code decoders find use in optical communication systems and LAN/WAN standards[10], [11]. Several hardware implementations of such decoders are commercially available[12], [13], [14]. Soft decision iterative decoders for product codes utilizing the "turbo-principle" have also been



(a) Relative error correction radii of the two algorithms are compared when decoding a 2D Product-Reed-Solomon code. The surface which dominates for most of the rate region corresponds to the new algorithm.



(b) 2D rate region where the new recursive algorithm performs better is shown.



(c) Relative rate region volume where our algorithm performs better is computed for various Product-Reed-Solomon codes of dimensionality m. The new algorithm is seen to out-perform the Pellikaan-Wu algorithm over much of the rate region.

Fig. 3: Comparison of Pellikaan-Wu algorithm to our new recursive RM/PRS decoder.

discussed in literature[5]. The performance of most of these hard decision iterative decoders can be very well characterized using Theorem 2. Similar conclusions are obvious for the case of other product codes which have algebraic bounded distance decoders available for their component codes. Theorem 2 implies the following for a general product code:

Corollary 1 *If for an m-dimensional product code $\mathbb{P}$, there exists bounded distance decoders for each of its component codes such that the i^{th} component code's decoder achieves a error correction radius of t_i errors, then there exists a decoding algorithm for the entire product code $\mathbb{P}$ which can correct all errors up to a weight of $t = \prod_{i=1}^{m} t_i$.*

The decoding algorithm for the code $\mathbb{P}$ mentioned in Corollary 1 can be obtained from Algorithm 2 with some obvious and minor changes and as such is not repeated here. This result is, to the best of the author's knowledge, the only such theoretical guarantee on the error correction radius of a general algebraic product code decoder. However for specific cases there are some stronger results available, for instance see the result of Lin-Weldon[1] for cyclic product codes. In another related example, Tanner[3] discusses bounds on a specific type of hard-decision decoder for product codes on graphs. In many cases of practical interest such bounds are difficult to apply because of their dependence on the knowledge of the girth of the underlying code graph.

IV. Conclusions

In this paper, we presented a simple and easily accessible proof for the Pellikaan-Wu algebraic list decoding algorithm for Reed-Muller codes. Our proof uses only the most fundamental properties of finite field arithmetic.

We also proposed a low complexity recursive algorithm for Reed-Muller and Product-Reed-Solomon codes. This new recursive algebraic decoding algorithm is then shown to have a significantly better error correction radius than the Pellikaan-Wu algorithm over a wide range of code rates.

References

[1] S. Lin and E. J. Weldon, "Further Results on Cyclic Product Codes," *IEEE Trans. Inform. Theory*, **16**, No. 4, pp. 452-459, Jul. 1970.

[2] J. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, **27**, No. 4, pp. 701 - 717, Oct. 1980.

[3] "A Recursive Approach to Low Complexity Codes," R. M. Tanner, *IEEE Trans. Inform. Theory*, **27**, No. 5, pp. 533-547, Sep. 1981.

[4] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, University of Cambridge Press, Cambridge, 1986.

[5] R. Pyndiah, "Near-Optimum Decoding of Product Codes: Block Turbo Codes," *IEEE Trans. Comm.*, **46**, pp. 1003-1010, Aug. 1998.

[6] V. Guruswami and M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Trans. Inform. Theory*, **45**, No. 6, pp. 1757-1767, Sep. 1999.

[7] R. Pellikaan and X.-W. Wu, "List Decoding of q-ary Reed-Muller Codes," *IEEE Trans. Inform. Theory*, **50**, No. 4, pp. 679-682, Apr. 2004.

[8] R. Pellikaan and X.-W. Wu, "List Decoding of q-ary Reed-Muller Codes," Expanded version of [7], manuscript available at `http://www.win.tue.nl/~ruudp/paper/43-exp.pdf`, Nov. 2005.

[9] F. Parvaresh, M. El-Khamy, R. J. McEliece and A. Vardy, "Algebraic List-decoding of Reed-Solomon Product Codes," Unpublished note of Jan. 2006, private communication.

[10] IEEE 802 LAN/MAN Standards Committee. Website at `http://www.ieee802.org/`

[11] The IEEE 802.16 Working Group on Broadband Wireless Access Standards. Website at `http://www.ieee802.org/16/`

[12] "AHA4540B 155 Mbits/sec TPC Encoder/Decoder IC," specifications available at `http://www.aha.com/show_prod.php?id=21`

[13] "IEEE 802.16 Compatible Turbo Product Code Decoder v1.1," specifications available at `http://www.xilinx.com/ipcenter/catalog/logicore/docs/tpc_decoder.pdf`

[14] "Very High-Speed Turbo Product Code Decoder TC3404," specifications available at `http://www.altera.com/products/ip/dsp/error_detection_correction/m-tur-tc3404.html`